Machine Learning, 50, 303–319, 2003

Clustered Partial Linear Regression

LUIS TORGO ltorgo@liacc.up.pt
LIACC-FEP, University of Porto, R. Campo Alegre 823, 2o., 4150 Porto, Portugal

JOAQUIM PINTO DA COSTA jpcosta@liacc.up.pt
LIACC-DMA, University of Porto, R. Campo Alegre 823, 2o., 4150 Porto, Portugal

Editors: Ryszard S. Michalski and Pavel Brazdil

Abstract. This paper presents a new method that deals with a supervised learning task usually known as multiple regression. The main distinguishing feature of our technique is the use of a multistrategy approach to this learning task. We use a clustering method to form sub-sets of the training data before the actual regression modeling takes place. This pre-clustering stage creates several training sub-samples containing cases that are "nearby" to each other from the perspective of the multidimensional input space. Supervised learning within each of these sub-samples is easier and more accurate as our experiments show. We call the resulting method clustered partial linear regression. Predictions using these models are preceded by a cluster membership query for each test case. The cluster membership probability of a test case is used as a weight in an averaging process that calculates the final prediction. This averaging process involves the predictions of the regression models associated to the clusters for which the test case may belong. We have tested this general multistrategy approach using several regression techniques and we have observed significant accuracy gains in several data sets. We have also compared our method to bagging that also uses an averaging process to obtain predictions. This experiment showed that the two methods are significantly different. Finally, we present a comparison of our method with several state-of-the-art regression methods showing its competitiveness.

Keywords: multistrategy learning, regression, clustering, multiple models

1. Introduction

This paper describes clustered partial linear models. This is a new method for addressing multiple regression problems. Multiple regression is a supervised learning task that can be loosely defined as the search for a model of the relationship between a target continuous variable and a set of other input variables. The technique we describe in this paper deals with this problem using a multistrategy learning approach.

Typical multistrategy learners integrate two or more inferential strategies in order to be able to solve more complex learning tasks (c.f. Michalski & Tecuci, 1994). This was not our aim when adopting a multistrategy framework as our learning task is a simple one. We have adopted a multistrategy approach with the goal of improving the predictive accuracy of our learned models. Our CPLR system uses both unsupervised and supervised learning. CPLR changes the initial learning task into a set of simpler sub-tasks by means of a clustering method. For each of these sub-tasks different models are obtained using a supervised learning method. During prediction a similar two-stage approach is followed. In a first step the clustering is used to predict the type of sub-task to which each test case belongs. This

information is then used to select the model (or models) that will make the final prediction for the given test case. As we will see the resulting multistrategy methodology has strong relations with multiple models approaches and also with standard partition-based methods, such as regression trees. Still, we will also show that the distinguishing features of our approach lead to highly significant predictive accuracy advantages in several regression domains.

Our general two-stage schema can be applied to any multiple regression method (and even to other supervised learning tasks). Still, in this paper we concentrate our description on partial linear regression models (Hardle, 1990; Spiegelman, 1976). However, we also report some results using other regression techniques.

This paper is organized as follows. The next section describes partial linear regression that is the basic technique that we use within our multistrategy approach to regression. Section 3 presents clustered partial linear models. In Section 4 we carry out an experimental analysis of these models. Section 5 discusses some approaches that are related to our work. Finally, we present the main conclusions of this work.

2. Partial linear models

Partial linear models (Hardle, 1990; Spiegelman, 1976) belong to the class of semi-parametric approaches that integrate parametric with non-parametric techniques. In the case of partial linear models, a standard least squares linear polynomial (e.g. Drapper & Smith, 1981) is integrated with a kernel smoother (Nadaraya, 1964; Watson, 1964). The main motivation behind these models is to retain as much as possible the comprehensibility of linear polynomials, while trying to improve their accuracy by adding a smoothing component that compensates, on a query-base, for the local inadequacies of the linearity assumption of first order polynomials.

A prediction for a query case using these models is obtained by combining the value predicted by the linear polynomial with the value resulting from smoothing over the residuals (errors) of the linear model in the neighboring training points. The more inadequate the linear model is to the given training sample the larger the importance of the smoothing component. In the extreme case where the linear component perfectly fits the training data, a partial linear model reduces to a standard least squares linear polynomial.

Given a data set, $\{\langle \mathbf{x}_i, y_i \rangle\}_{i=1}^{n}$, where $\mathbf{x}_i$ is a vector of continuous attribute values, a multiple linear regression model of the form $Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n$, can be obtained using a least squares error criterion. This consists of finding the vector of parameters β that minimizes the sum of the squared errors, i.e. $(\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta)$, where $\mathbf{X}'$ denotes the transpose of matrix $\mathbf{X}$. After some matrix algebra the minimization of this expression with respect to β leads to the following set of equations, usually referred to as the *normal equations* (e.g. Drapper & Smith, 1981),

$$(\mathbf{X}'\mathbf{X})\beta = \mathbf{X}'\mathbf{y} \tag{1}$$

The parameter values can be obtained by solving the equation,

$$\beta = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \tag{2}$$

where $\mathbf{X}^{-1}$ denotes the inverse of matrix $\mathbf{X}$.

As the inverse matrix does not always exist this process suffers from numerical instability. A better alternative (Press et al., 1992) is to use a set of techniques known as Singular Value Decomposition (SVD), that can be used to find solutions of systems of equations with the form $\mathbf{X}\beta = \mathbf{y}$.

A kernel smoother (Nadaraya, 1964; Watson, 1964) can be seen as a form of lazy learner (Aha, 1997) that delays learning until prediction time. Given a query point $\mathbf{x}_q$, a prediction is obtained using the following expression,

$$y'_q = \frac{1}{\text{SKs}} \sum_{i=1}^{n} K\left(\frac{d(\mathbf{x}_i, \mathbf{x}_q)}{h}\right) \times y_i \tag{3}$$

where, $d(\cdot)$ is the distance function between two instances; $K(\cdot)$ is a kernel function; h is a bandwidth value; $\langle \mathbf{x}_i, y_i \rangle$ is a training instance; and SKs is the sum of the weights of all training cases, i.e. $\text{SKs} = \sum_{i=1}^{n} K(\frac{d(\mathbf{x}_i, \mathbf{x}_q)}{h})$.

This formula is a weighed average over the target values of the training cases that are nearer to the query point. The notion of neighborhood implies the definition of a metric over the multi-dimensional space defined by the input variables and of a distance function between any two cases in this space. The bandwidth, h, provides a scaling effect of the cases within the neighborhood that *enters* the weighed average. The kernel function, $K(\cdot)$, provides a smoothing effect, giving more *importance* to nearer training cases. Many different variants of all these parameters of kernel smoothing are described in the literature (e.g. Atkeson, Moore, & Schaal, 1997).

Partial linear models integrate a linear polynomial with a kernel smoother applied on the residuals (errors) of the polynomial. The role of the kernel smoother is to provide an estimate of the error of the linear polynomial for the particular query case under consideration. This estimated error is then subtracted from the linear polynomial prediction giving the predicted value of the partial linear model. Formally, this can be described by,

$$y'_q = \beta \mathbf{x}_q - \frac{1}{\text{SKs}} \sum_{i=1}^{n} K\left(\frac{d(\mathbf{x}_i, \mathbf{x}_q)}{h}\right) \times e_i \tag{4}$$

where, e_i is the error of the linear model in case $\langle \mathbf{x}_i, y_i \rangle$, given by $e_i = \beta \mathbf{x}_i - y_i$.

Thus, to obtain a prediction for a query case using a partial linear model, we start by obtaining the predicted value of the linear polynomial, $\beta \mathbf{x}_q$. We then calculate the error of this linear polynomial in the training cases that are nearer to the query point, $\mathbf{x}_q$. Using these errors we obtain a kernel prediction of the error for the query case. Finally, this predicted error is subtracted from the initial value predicted by the linear polynomial giving the prediction of the partial linear model.

Compared to linear polynomials, partial linear models have significant advantages in terms of predictive accuracy when the domain is non-linear. Contrary to kernel smoothers, partial linear models have some degree of comprehensibility due to the use of a linear polynomial. However, as they also incorporate a kernel component (that is not comprehensible), they are less understandable than linear regression. In effect, the polynomial component of partial linear models can only be regarded as a rough description of the true surface

approximated by these models. The accuracy of this description is proportional to the linearity of the domain under study. In highly non-linear domains the predominance of the kernel *corrections* is so high that the linear polynomial is a very poor *explanation* of the predictions of the partial linear model (Torgo, 2000).

3. Clustered partial linear models

Lazy learners have witnessed a noticeable success in many application domains (e.g. Atkeson, Moore, & Schaal, 1997; Bontempi, 2000). The models presented in Section 2 also belong to this family of techniques. As we have seen, lazy learners perform a kind of local modeling[1] for each query case. In effect, given a query case these approaches start by finding the training instances that are *nearer* in terms of the input variables to this query point. Using only these observations lazy learners obtain a local model that can be used to obtain a prediction for this particular query point. Different models can be fit to these local neighborhoods (Cleveland & Loader, 1995), such as polynomials of degree zero (as in kernel regression and k-NN), or higher order polynomials (like in local linear regression). In spite of their success these techniques have some drawbacks. Among them the most noticeable are the fact that the neighborhoods are obtained on a query by query basis (which is computationally demanding), and also the fact that these techniques do not produce a visible and interpretable model of the training data.

The regression method we propose is motivated by this analysis of local modeling. We have tried to develop a method that obtains models for sub-sets of *nearby* training cases avoiding the computational drawbacks of lazy learners by using a fixed set of these *interesting neighborhoods*, instead of obtaining them for each query case. This idea is implemented in system CPLR that learns clustered partial linear models.

3.1. Learning clustered partial linear models

Clustered partial linear models are learned in two sequential stages. In the first stage we try to obtain a set of *neighborhoods* of training cases. As in lazy learners these neighborhoods are found by looking only at the input variables (i.e. irrespectively of the target variable value of the cases). This means that we are searching for groups of training cases that are nearby to each other within the multidimensional input space. To accomplish this task we have used a clustering algorithm. This clustering stage is followed by a supervised learning step that obtains a regression model for each found cluster.

We have used AUTOCLASS C (Cheesman et al., 1988; Cheesman & Stutz, 1995) to obtain the initial clusters of training cases. AUTOCLASS C automatically chooses the number of clusters and associates cluster membership probabilities to each training case. We use this latter feature to add some flexibility to the cluster bounds. Instead of allocating each training case to the cluster for which AUTOCLASS C predicts a higher membership probability, we allocate it to all clusters for which there is a membership probability higher then zero. This means that as a result of our initial clustering stage we obtain a set of training sub-samples (the clusters obtained by AUTOCLASS C) with some degree of overlap, induced by the cluster membership probabilities of AUTOCLASS C.

After this clustering-based partitioning of the training data the second stage of our multistrategy approach consists of a supervised learning step. Namely, we fit a partial linear regression model to each of the obtained training sub-samples. This consists of obtaining a least squares linear polynomial for each cluster, because the kernel component does not involve any *training* (c.f. Section 2). In summary, clustered partial linear models consist of a set of partial linear models built for different clusters of the data.

If the used clustering algorithm is able to produce a symbolic representation of each cluster it is possible to consider clustered partial linear models as a single model. In effect, the symbolic representation of each cluster can be seen as a pre-condition for applying the respective partial linear model, which means that we can look at a clustered partial linear model as a set of rules of the form,

```
IF <cluster representation> THEN <partial linear model>.
```

Even if the clustering algorithm does not produce such symbolic representation of the clusters, we could use any classification algorithm to produce such description, in the style of ABACUS (Falkenhainer & Michalski, 1990). In effect, the instances within each cluster can be labelled as instances of a class and a standard classification system can be used to obtain a symbolic description of each class (cluster).

3.2. *Predictions using clustered partial linear models*

Regarding predictions using clustered partial linear models they are obtained as follows. Given a query case $\mathbf{x}_q$, we use AUTOCLASS C to obtain the probabilities of the case belonging to each of the clusters. For each cluster with membership probability higher than zero, the respective partial linear model is used to obtain a prediction for the query case. These predictions are then averaged to obtain the final predicted value using the formula,

$$\text{CPLR}(\mathbf{x}_q) = \sum_{k=1}^{J} (\text{P}_k(\mathbf{x}_q)\text{pl}_k(\mathbf{x}_q)) \tag{5}$$

where, J is the number of clusters; $\text{P}_k(\mathbf{x}_q)$ is the probability that the query case belongs to cluster k; and $\text{pl}_k(\mathbf{x}_q)$ is the prediction of the partial linear model of cluster k (Eq. (4)) for the query case.

3.3. *An illustrative example*

This section describes an example of running CPLR on a particular domain in order to provide a better intuition of its behavior. We have used the *Abalone* data set, which concerns the task of trying to predict the number of rings in the shells of abalones (which is related to their age) based on a series of biometrics of these animals (see Table 2 for further details). Running AUTOCLASS C on the 4177 available training cases we obtain 8 clusters with 1434, 1336, 1049, 152, 100, 88, 13 and 5 cases, respectively. These numbers are obtained by assigning each case to its most probable cluster. As AUTOCLASS C may also give lower

probabilities that a case belongs to other clusters, and in these situations we have decided to allocate the same instance to all the alternative clusters (c.f. Section 3.1), the final 8 corresponding training sub-samples have 2690, 3408, 3584, 2351, 1989, 515, 55 and 10 cases, which provides an idea of the degree of overlap between the 8 training sets. The next step is to develop a partial linear model for each of these training sets. As a partial linear model includes a non-symbolic component (the kernel) we can only show some examples of part of the models (i.e. the linear polynomial component of the partial linear model). In spite of this lack of descriptive precision we can try to compare some of the models obtained for the 8 different training sets, and the model that would be obtained if the clustering phase was not carried out (i.e. applying a partial linear model to all training data). For instance, for the first training set (2690 cases) the linear model is:

$$N.Rings = 6.24 + 15.4 \times Height + 9.49 \times WholeWeight - 18 \times ShuckedWeight - 9.88 \times VisceraWeight + 8.75 \times ShellWeight$$

For the 6th training set (515 cases) we get:

$$N.Rings = 1.33 - 29.9 \times ShuckedWeight + 21.1 \times Diameter + 14.1 \times WholeWeight$$

Finally, if we used all training data (4177 cases) we would get the following single model:

$$N.Rings = 2.9 + 11.6 \times Diameter + 11.8 \times Height + 9.26 \times WholeWeight - 20.3 \times ShuckedWeight - 9.93 \times VisceraWeight + 8.61 \times ShellWeight$$

The diversity of the models is a good indication of the differences between the found clusters and the full training sample. This was also confirmed by some of the statistics output by AUTOCLASS C, like the cross entropy between each cluster and the full training data, that gives a measure of the difference between two probability distributions (higher values indicate further divergence from the probability distribution of all training data). These statistics are shown in Table 1.

Table 1. Divergence of the obtained clusters.

Cluster	Divergence value
0	2.48e+00
1	1.81e+00
2	1.21e+00
3	4.11e+00
4	7.53e+00
5	1.30e+01
6	1.13e+02
7	6.45e+02

AUTOCLASS C outputs a series of statistics that give an idea of which input variables better describe each cluster. Still, this can hardly be considered a symbolic and easily interpretable description of each cluster. Because of this we have tried to use C4.5RULES (Quinlan, 1993) to produce such cluster descriptions following the idea outlined in the end of Section 3.1. C4.5RULES generated 12 rules to describe the 8 different classes. For each class (cluster) different rules may be generated. If we join the rules of each class we get a description of the respective cluster. This is obviously an approximate description as C4.5RULES produces axes-parallel partitions of the input space, which is not the case of the clusters obtained by AUTOCLASS C. As an example, for the 6th training set (cluster) we would obtain the rule,

IF

$Sex = \text{I}$ AND

(*ShellWeight* ≤ 0.04 OR (*ShellWeight* ≤ 0.11 AND *ShuckedWeigth* ≤ 0.06))

THEN

$N.Rings = 1.33 - 29.9 \times ShuckedWeight + 21.1 \times Diameter + 14.1 \times WholeWeight.$

This interpretable description of the 6th cluster indicates that AUTOCLASS C formed a cluster using training cases that represent infant ($Sex = \text{I}$) abalones with small weight. Using this strategy we can obtain an interpretable overview of the model learned by CPLR.

In summary, this example shows how our multistrategy approach changes a regression problem into J different regression sub-problems, that are quite different from the initial task. The clustering phase is in effect performing a kind of *informed partitioning* of the data, that creates several sub-problems with quite different probability distributions from the original data. As we will show in Section 4 this clustering-based partitioning of the data is advantageous in terms of predictive accuracy.

4. Experimental analysis

In this section we describe a series of experiments with CPLR in order to assess its performance when compared to other related techniques, and also to provide a better understanding of its limitations. Most of the experiments are carried out using the data sets described in Table 2. With respect to the experimental methodology all reported results are averages of five repetitions of a 10-fold Cross Validation experiment. Significances of the differences in mean squared error (MSE) are asserted through paired t-tests.

4.1. Clustered versus non-clustered regression

The first experiment we report was designed to check whether there is a significant difference in accuracy between applying regression models in clustered training sub-samples (our CPLR proposal) or to all available training data.

Table 2. The used data sets.

Data set	Main characteristics
Housing	506 case; 13 continuous variables. Predicting housing values in Boston.
Abalone	4177 cases; 7 cont. vars.; 1 nominal var. Predicting the age of abalone.
Elevators	8752 cases; 40 cont. vars. Aircraft control problem (prediction of elevators level).
F	3000 cases; 5 cont. vars. Artificial domain with marked clusters of data points.
Kinematics	8192 cases; 8 cont. vars. Robot arm control problem.
Computer	8192 cases; 22 cont. vars. Prediction of CPU activity level in a computer network.
Computer (small)	8192 cases; 12 cont. vars. Simplified version of the previous data set.
Telecomm	15000 cases; 49 cont. vars. Telecommunications problem.

Table 3. The advantage of clustering the data.

	Partial linear models			Regression trees		
Data set	Clustered	All data		Clustered	All data	
Housing	13.04	16.94	+	25.42	20.13	
Abalone	2.28	4.75	+	5.55	5.41	
Elevators	5.61	15.07	+	9.73	17.38	+
F	1.56	3.45	+	4.31	7.56	+
Kinematics	0.010	0.012	+	0.032	0.039	+
Computer	24.38	22.07	−	8.21	12.35	+
Computer (small)	13.98	13.93	−	7.65	14.36	+
Telecomm	89.86	53.72	−	76.71	61.90	−

We have tested our clustering-based method with three different types of regression models: partial linear models; multiple linear regression models (as described in Section 2, Eqs. (1) and (2)); and regression trees (e.g., Breiman et al., 1984 or Torgo, 1999). For each of these trials we have compared the *clustered* variant, obtained using the methodology described in Section 3, with the alternative of simply applying the regression model to all available training data. The results for two of these regression models are shown in Table 3. Significant wins (99% confidence) of the *clustered* versions are marked with + signs, while significant wins of the *unclustered* versions appear with − signs. Clustered multiple linear regression models achieved significant wins on all data sets (c.f. Torgo & Costa, 2000),

although in this case the goal of outperforming the model obtained with all training data is easier because of the strong assumptions of multiple linear regression.

These experiments confirm the advantages of pre-clustering the data as we propose. In effect, with all three regression methods (that are quite different) there is a general trend towards a significant gain in predictive accuracy. However, the results of clustered partial linear models on the two *Computer* domains and in the *Telecomm* application are a bit disappointing. A possible cause of these results is the complete inadequacy of linear polynomials to these domains, which was confirmed in the results presented in the above mentioned paper by Torgo and Costa. Although partial linear models include a smoothing component that could overcome this mismatch, there are situations were this is not possible. In effect, the lack of symmetry near the boundaries of the input space causes well known difficulties to kernel smoothers (Hastie & Loader, 1993). The extrapolation capabilities of linear polynomials may also lead to *wild* predictions near these boundaries. These two factors together may explain the poor performance on these domains. This explanation is consistent with the fact that clustered regression trees (that do not have such difficulties) do not achieve such disappointing results. Thus, we claim that these poor results are caused by the lack of adequacy of the base regression models to the domains and not by any difficulty of our proposed multistrategy methodology. In effect, these results provide a motivation for a possible extension of CPLR in order to explore the possibility of using different regression models for each cluster.

4.2. *Comparison to approaches using multiple models*

Clustering the given training data is just one of the distinguishing features of our methodology. Another difference from standard non-clustered regression methods, is the averaging of the predictions of different models. Regarding this issue our methodology resembles bagging (Breiman, 1996), were predictions are obtained by averaging over a set of models learned using different bootstrap samples of the given data. According to Breiman (1996) bagging is expected to give good results when the base prediction method is sensible to small perturbations on the learning set, as it is the case for regression trees. However, that is not the case of partial linear models that are quite robust to these variations, as the results of Table 4 confirm. This table shows the results of a comparison between partial linear models and regression trees with their respective *bagged* versions. These results confirm Breiman's statement. The *bagged* models were obtained using 50 bootstrap replicates of the data. Significant wins of the bagged versions are marked with + signs.

The results of these experiments with bagging show that the accuracy advantages of clustered partial linear models that were observed in Table 3, are mainly caused by the effects of clustering the training data. In effect, the results of Table 4 indicate that there is nothing to gain with averaging over several partial linear models.

4.3. CPLR *versus standard partition-based regression*

The clustering phase of our CPLR models can be seen as a form of obtaining sub-sets of the data for which different models are built afterwards. From this perspective, CPLR is related

Table 4. Bagging regression models.

	Partial linear models			Regression trees		
Data set	Single	Bagged		Single	Bagged	
Housing	16.94	17.12		20.13	13.02	+
Abalone	4.75	4.68	+	5.41	4.63	+
Elevators	15.07	15.11		17.38	10.18	+
F	3.45	3.41		7.56	4.04	+
Kinematics	0.012	0.012		0.039	0.024	+
Computer	22.07	22.05		12.35	8.20	+
Computer (small)	13.93	14.14	−	14.36	9.75	+
Telecomm	53.72	54.56	−	61.90	37.08	+

to standard partition-based methods, such as regression trees. However, the way these two approaches build the sub-sets is totally different. In effect, while CPLR uses the information on the input variables to form the sets of cases, typical partition-based systems use only the information on the target variable to divide the training cases into partitions. For instance, standard least squares regression trees (e.g. Breiman et al., 1984) search for partitions with low variance in the target variable. In this section we address the question of whether this difference is relevant in terms of predictive accuracy.

CPLR is particularly related to partial linear trees (Torgo, 2000) that consist of standard regression trees with partial linear models in the leaves instead of the usual average values. In effect, the single difference between these two approaches is on the method used to obtain sub-sets of training cases, as they share the code that implements partial linear models. We have compared CPLR to partial linear trees (PLT) in our benchmark data sets. Table 5 shows the results of this experimental comparison. Significant wins of clustered partial linear models are marked with + signs.

The results of this experiment confirm that the way CPLR obtains sub-sets of the given training data is clearly different from typical partition-based methods. Moreover, the way

Table 5. Clustered partial linear regression (CPLR) versus partial linear trees (PLT).

Data set	CPLR	PLT	
Housing	13.04	17.28	
Abalone	2.28	5.31	+
Elevators	5.61	9.94	+
F	1.56	4.50	+
Kinematics	0.010	0.028	+
Computer	24.38	8.21	−
Computer (small)	13.98	11.05	−
Telecomm	89.86	44.77	−

Table 6. Clustered partial linear regression (CPLR) versus other multiple regression approaches.

Data set	CPLR	Cubist		Mars		Bagged CART	
Housing	13.04	14.24		18.32	+	13.02	
Abalone	2.28	4.67	+	4.54	+	4.63	+
Elevators	5.61	12.91	+	6.04		10.18	+
F	1.56	13.64	+	19.95	+	4.04	+
Kinematics	0.010	0.027	+	0.036	+	0.024	+
Computer	24.38	6.49	−	10.22	−	8.20	−
Computer (small)	13.98	9.71	−	14.00		9.75	−
Telecomm	89.86	91.51		??[2]		37.08	−

CPLR proceeds seems to provide significant predictive accuracy advantages on several data sets, although the opposite also occured in other domains. Still, we have already seen in Section 4.1 that these poor results are caused by the models used in the sub-sets and not by the method used to form the clusters.

4.4. CPLR *versus other multiple regression approaches*

Having shown the advantages of a pre-clustering of the training sample, it remains an open question whether the results of clustered partial linear models are good when compared to other existing approaches to multiple regression. We have compared CPLR with several state-of-the-art regression systems, namely, CUBIST (http://www.rule-quest.com), MARS (Friedman, 1991) and a bagged version of CART (Breiman et al., 1984). The results of this experiment are shown in Table 6. Significant wins of clustered partial linear models are marked with + signs.

CPLR achieves quite competitive accuracy in most domains. Some results are particularly outstanding, namely in the *Abalone*, *F* and *Kinematics* domains. Moreover, in the case of *Abalone* and *F*, the excellent scores are clearly caused by the clustering step because neither the single models nor the *bagged* versions obtain similar results (c.f. Tables 3 and 4). As we have mentioned in Section 4.1 the less encouraging results on the two *Computer* domains and in the *Telecomm* data set are partially explained by the high non-linearity of the domains and could possibly be overcome using other regression models in the found clusters.

4.5. *Lesion experiments*

This section describes a series of experiments with artificial data sets with the goal of providing a better understanding of the limitations of CPLR. Namely, we have carried out some experiments with artificial domains to understand how the existence of marked clusters of cases in the training data would influence the predictive accuracy of CPLR.

We have used three artificial data sets with two input variables to allow the graphical presentation of the results. The data sets are formed by three groups of 100 observations in a total of 300 cases. Within each of the three groups the target variable value is obtained

using the following three functions,

- 1st group

$$y = 10 \sin(\pi x_2) + 20 (x_1 - 0.5)^{10} + \sigma(0, 1)$$

- 2nd group

$$y = 10 + \pi x_1^3 + \sigma(0, 1)$$

- 3rd group

$$y = 0.8 - 10 \sin(x_2 (-x_1)) + \sigma(0, 1)$$

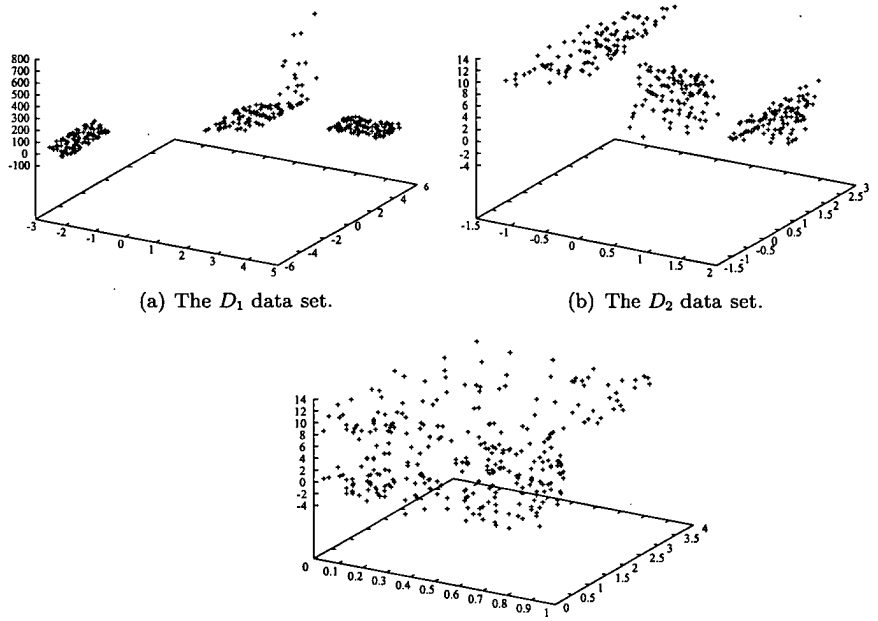
where, $\sigma(0, 1)$ is a gaussian noise term and x_1 and x_2 are the values of the two input variables. The three data sets differ in the generation of the values of the input variables x_1 and x_2. Namely, for each data set we have forced a different spread in terms of the input space, of the three groups of 100 cases. Figure 1 shows the three data sets. The first data set, D_1, shown on figure 1(a), has three very marked clusters of points. The D_2 data set (figure 1(b)) has less marked clusters, while D_3 (figure 1(c)) has a strong overlap between the three groups of 100 observations.

Figure 2 shows the training sub-samples obtained by CPLR for each data set. CPLR is able to capture almost perfectly the original clusters of the data (c.f. figure 1) for data sets D_1 and D_2. For data set D_3 CPLR "invents" 7 clusters of data.

We have compared CPLR with the alternative of fitting partial linear models on all data (as in Table 3) to understand the influence of the "natural" clusters in the data on the predictive accuracy of CPLR. The results of this experiment are shown in Table 7. Significant accuracy advantages of CPLR are marked with + signs, and the average number of used clusters are shown on the second line between parentheses.

As expected the advantage of CPLR is related to the existence of marked clusters of observations in the data. However, even when that does not happen the performance of CPLR was quite competitive on these artificial domains.

In spite of the promising accuracy results our methodology also has some disadvantages. The most noticeable is the increment of computation time when compared to applying a regression model to all training data. This additional cost is caused by the clustering step. AUTOCLASS C has many parameters that could be explored in order to try to improve this speed issue. Alternatively, we could explore other fast clustering techniques like the ones describe by Bradley, Fayyad, and Reina (1999) or Farnstrom, Lewis, and Elkan (2000). Still, clustering is always a heavy task and its weight in the overall computation time of clustered partial linear models will always be high. However, when compared to standard lazy learners we expect CPLR to be quite competitive in terms of testing time. In effect, while lazy learners will search for the neighbors of a test case in all data set, CPLR will only carry out this search in sub-sets of the training data, which may be extremely beneficial particularly if the training sample is very large. Obviously, this difference will only be relevant if one has to make predictions for a large amount of test cases. Only with such



(a) The D_1 data set.

(b) The D_2 data set.

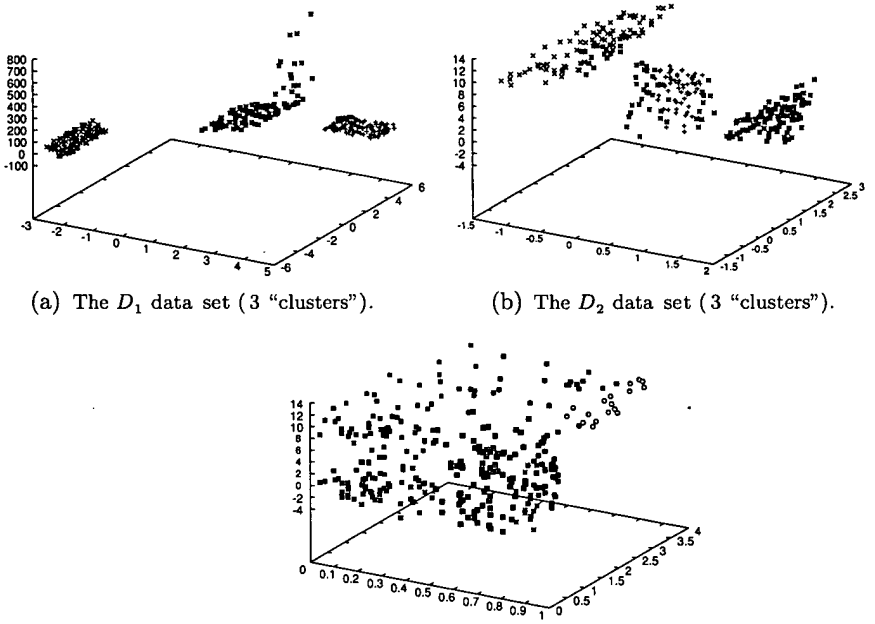
(c) The D_3 data set.

Figure 1. The three artificial domains.

large test sets could the gains in testing time compensate for the losses in training time of CPLR with respect to lazy learners.

5. Related work

As we have already mentioned there are some relations of our method with multiple model approaches like bagging (Breiman, 1996). Both approaches obtain different models based on possibly overlapping samples of cases and perform an averaging of these models as a means of obtaining predictions for test cases. However, the method we use to obtain the individual samples is totally different. In bagging a bootstrap random sampling process is used to obtain samples with the same size as the original training sample. In our methodology the samples are not obtained randomly and are usually smaller than the original training sample. The type of partitioning carried out by our clustering step changes the distribution of the cases in the original sample, which is not the case with bagging. From this perspective, our method is related to boosting (Shapire, 1990; Freund & Shapire, 1996), where a distribution change is carried out through a system of weights. However, contrary to boosting our method is not sequential and thus it is possible to construct the individual models in parallel.



(a) The D_1 data set (3 "clusters").

(b) The D_2 data set (3 "clusters").

(c) The D_3 data set (7 "clusters").

Figure 2. The training sub-samples created by CPLR on the three data sets.

ABACUS (Falkenhainer & Michalski, 1990) is a multistrategy system that can also be related to our approach although it follows a different methodology. ABACUS starts by using an equation discovery method to derive equations that can summarize the given data. If more than one equation is necessary the original data is divided accordingly. After this step a classification system is used to discriminate among the obtained divisions of the data. The found descriptions can be regarded as pre-conditions for the application of the previously found equations. Our system, on the contrary, starts by finding relevant clusters

Table 7. The performance of CPLR on the artificial data sets.

Data set	CPLR	All data	
D_1	3116.9	4059.3	+
	(3.4)		
D_2	1.63	1.93	+
	(3.2)		
D_3	15.34	15.55	
	(7.5)		

of the data irrespectively of the models that are going to be used, and only in a second stage models are obtained for each cluster. In effect, a similar approach to ours is slightly outlined as a future research direction in the above mentioned paper by Falkenhainer and Michalski. Still, other distinguishing features of our work include the use of overlapping clusters and also the use of a probabilistic cluster membership function when making predictions about query cases, which leads to the combination of the values predicted by different models.

Devogelaere, Bael, and Rijckaert (1999) describe a related approach to regression. Their GAdC system performs a genetic algorithm driven clustering of the training data. The evaluation function that drives the genetic algorithm-based search for the clusters includes several factors like cluster distance penalty, prediction error, etc. This means that, unlike our method, GAdC uses information about the regression accuracy to guide the search for clusters. Within each cluster, GAdC uses a kind of kernel model to obtain predictions. Another difference of our work is the probabilistic approach to cluster membership that leads to overlap of clusters and also to the averaging of different clustered models.

CPLR is related to typical partition-based learners, like tree-based models or rule-based systems. These methods partition the given sample in a set of local regions and fit some kind of model within each of these regions. However, there are some fundamental differences from our method. The most important is the criteria used to form the sub-sets of cases. Regression trees (e.g. Breiman et al., 1984), for instance, search for regions of low variance in the target variable. Our method does not use any information regarding the target variable to obtain the sub-sets. On the contrary these groups of cases are obtained based on information concerning the input variables in a lazy learning fashion. Other differences to these approaches are the possible overlap of the regions, the probabilistic approach to decide on which cluster to *place* a test case, and the averaging over different regions in prediction tasks.

Schulmeister and Wysotzki (1997) present DIPOL, a supervised classification system that has some features related to CPLR. DIPOL handles a different supervised learning task but it can also perform a pre-clustering of the training data in a similar fashion to CPLR. This step is motivated by the restrictions imposed by the decision regions induced by DIPOL. In particular, every decision region must be singly connected, which leads to difficulties to handle non-convex class regions (such as the XOR problem). In such cases, the authors propose to use a clustering of the data prior to the modeling stage in order to overcome these difficulties. Although with a different motivation this step resembles the initial stage of CPLR. Still, DIPOL uses a different clustering algorithm, it does not generate overlapping clusters and it does not perform model averaging, as CPLR does.

6. Conclusions

We have described a multistrategy learning approach to multiple regression whose main distinguishing feature is the use of a clustering algorithm to obtain sub-samples of the available training data. These samples are then modeled individually using partial linear regression models. Predictions using the resulting clustered partial linear models are obtained by averaging over the models of the clusters for which the membership probability of the test cases is higher than zero.

We have tried this clustering-based approach to regression with three different types of regression models. With all three methods we have observed a similar pattern of results, showing the advantages of pre-clustering the training data.

We have compared our method to bagging that also uses averaging over multiple models. The results show that there are significant differences between the two methods, with some clear advantages of our approach.

Compared to existing state of the art regression approaches our method achieved quite competitive results in the tested domains. The accuracy in some domains can be considered particularly outstanding.

Overall, CPLR can be considered a quite competitive multiple regression system in terms of predictive accuracy. However, this comes with some costs in terms of computation efficiency, particularly when compared to fast methods like regression trees.

Acknowledgments

This work is partially supported by project Sol-Eu-Net IST-1999-11495, by PRAXIS XXI plurianual support to LIACC, and by ESPRIT LTR Project METAL 26.357.

Notes

1. In effect, these approaches are also known as *local regression* (Cleveland & Loader, 1995) or *local modeling* (Fan, 1995) within statistics.
2. The version of MARS we use gives a segmentation fault on this data set.

References

Aha, D. (1997). Lazy learning. *Artificial Intelligence Review, 11*.

Atkeson, C., Moore, A., & Schaal, S. (1997). Locally weighted learning. *Artificial Intelligence Review, 11*, 11–73.

Bontempi, G. (2000). Local learning techniques for modeling, prediction and control. Ph.D. Thesis, Universit Libre de Bruxelles, Belgium.

Bradley, P., Fayyad, U., & Reina, C. (1999). Scaling clustering algorithms to large databases. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining* (pp. 9–15). AAAI Press.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*, 123–140. Aggregating predictions, prediction, bagging, combination methods.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and Regression Trees,* Statistics/Probability Series. Wadsworth & Brooks/Cole Advanced Books & Software.

Cheesman, P., Kelly, J., Self, M., & Stutz, J. (1988). Autoclass: A Bayesian classification system. In *Proceedings of the 5th International Conference on Machine Learning*. San Mateo, CA: Morgan Kaufmann.

Cheesman, P., & Stutz, J. (1995). Bayesian classification (Autoclass): Theory and Results. In *Advances in Knowledge Discovery*. AAAI Press.

Cleveland, W., & Loader, C. (1995). Smoothing by local regression: Principles and methods (with discussion). *Computational Statistics*.

Devogelaere, D., Bael, P. V., & Rijckaert, M. (1999). Regression through genetic algorithm driven clustering. In *Proceedings of the 7th European Congress on Intelligent Techniques and Soft Computing (EUFIT'99)*.

Drapper, N., & Smith, H. (1981). *Applied regression analysis,* 2nd ed. New York: John Wiley & Sons.

Falkenhainer, B., & Michalski, R. (1990). Integrating quantitative and qualitative discovery in the ABACUS system. In *Machine learning and artificial intelligence approach* (Vol. III). San Mateo, CA: Morgan Kaufmann.

Fan, J. (1995). Local modelling. In *Encyclopedia of statistical science*.

Farnstrom, F., Lewis, J., & Elkan, C. (2000). Scalability for clustering algorithms revisited. *SIGKDD Explorations, 2:1*, 51–57.

Freund, Y., & Shapire, R. (1996). Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning*. San Mateo, CA: Morgan Kaufmann.

Friedman, J. (1991). Multivariate adaptive regression splines. *The Annals of Statistics, 19:1*, 1–141.

Hardle, W. (1990). *Applied nonparametric regression*. Cambridge, UK: Cambridge University Press.

Hastie, T., & Loader, C. (1993). Local regression: Automatic kernel carpentry. *Statistical Science, 8*, 120–143.

Michalski, R., & Tecuci, G. (Eds.). (1994). *Machine Learning, a multistrategy approach* (Vol. IV). San Mateo, CA: Morgan Kaufmann.

Nadaraya, E. (1964). On estimating regression. *Theory of probability and its applications, 9*, 141–142.

Press, W., Teukolsky, S., Vetterling, W., & Flannery, B. (1992). *Numerical Receipes in C*. Cambridge, UK: Cambridge Univeristy Press.

Quinlan, J. (1993). *C4.5: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann Publishers.

Schulmeister, B., & Wysotzki, F. (1997). DIPOL—A hybrid piecewise linear classifier. In *Machine Learning and Statistics, the Interface*. New York: John Wiley & Sons, Inc.

Shapire, R. (1990). The strength of weak learnability. *Machine Learning, 5*, 197–227.

Spiegelman, C. (1976). Two techniques for estimating treatment effects in the presence of hidden variables: Adaptive regression and a solution to Reiersol problem. Ph.D. Thesis, Dept. of Mathematics, Nortwestern University.

Torgo, L. (1999). Inductive learning of tree-based regression models. Ph.D. Thesis, Faculty of Sciences, University of Porto.

Torgo, L. (2000). Partial linear trees. In P. Langley (Ed.), *Proceedings of the 17th International Conference on Machine Learning* (pp. 1007–1014). San Mateo, CA: Morgan Kaufmann.

Torgo, L., & Costa, J. P. (2000). Clustered partial linear regression. In R. L. de Mantaras, & E. Plaza (Eds.), *Proceedings of the 11th European Conference on Machine Learning (ECML 2000)*, number 1810 in LNAI, (pp. 426–436). Berlin: Springer.

Watson, G. (1964). Smooth regression analysis. *Sankhya: The Indian Journal of Statistics*, Series *A 26*, 359–372.

Received October 20, 2000
Revised September 12, 2001
Accepted May 17, 2002
Final manuscript June 20, 2002